# SNMP

## SNMP简介

Simple Network Management Protocol 简单网络管理协议

SNMP指的是简单网络管理协议。它属于TCP/IP五层协议中的应用层协议。它提供了一种简单和方便的模式来管理网络中的各个元素。这里的元素就是各个被管理的对象，可以是因特网中的某个硬件，比如网卡，也可以是某些硬件和软件的配置参数的集合。由于SNMP协议简单可靠 ，受到了众多厂商的欢迎，成为了目前最为广泛的网管协议。

SNMP协议主要由两大部分构成：SNMP管理站和SNMP代理。SNMP管理站是一个中心节点，负责收集维护各个SNMP元素的信息，并对这些信息进行处理，最后反馈给网络管理员；而SNMP代理是运行在各个被管理的网络节点之上，负责统计该节点的各项信息，并且负责与SNMP管理站交互，接收并执行管理站的命令，上传各种本地的网络信息。

SNMP管理站和SNMP代理之间是松散耦合。他们之间的通信是通过UDP协议完成的。一般情况下，SNMP管理站通过UDP协议向SNMP代理发送各种命令，当SNMP代理收到命令后，返回SNMP管理站需要的参数。但是当SNMP代理检测到网络元素异常的时候，也可以主动向SNMP管理站发送消息，通告当前异常状况。

SNMP协议于1988年发布。到目前一共经历了V1，V2，V3三个版本。其中V1已经被废弃，而V2c虽然没有能够成为正式标准，，但是已经被很多厂家所接受，V3目前是因特网的正式标准。与V1相比，V2，V3更能适应大规模的网络管理，而且在安全方面有了较大的改进。

# SNMP4J

http://www.xuebuyuan.com/2163204.html

## 重要的类和接口

Snmp类：该类是SNMP4J中最为核心的类。负责SNMP报文的接受和发送。

PDU类和ScopedPDU类：该类是SNMP报文单元的抽象，其中PDU类适用于SNMPv1和SNMPv2c。ScopedPDU类继承于PDU类，适用于SNMPv3。

Target接口和UserTarget类：对应于SNMP代理的地址信息，包括IP地址和端口号（161）。其中Target接口适用于SNMPv1和SNMPv2c。UserTarget类实现了Target接口，适用于SNMPv3。

TransportMapping接口：该接口代表了SNMP4J所使用的传输层协议。这也是SNMP4J一大特色的地方。按照RFC的规定，SNMP是只使用UDP作为传输层协议的。而SNMP4J支持管理端和代理端使用UDP或者TCP进行传输。该接口有两个子接口。

## 两种消息发送模式

SNMP4J支持两种消息发送模式：同步发送模式和异步发送模式。

其中同步发送模式也称阻塞模式。当管理端发送出一条消息之后，线程会被阻塞，直到收到对方的回应或者时间超时。同步发送模式编程较为简单，但是不适用于发送广播消息。

异步发送模式也称非阻塞模式。当程序发送一条消息之后，线程将会继续执行，当收到消息的回应的时候，程序会对消息作出相应的处理。要实现异步发送模式，需要实例化一个实现了ResponseListener接口的类的对象。ResponseListener接口中有一个名为onResponse的函数。这是一个回调函数，当程序收到响应的时候，会自动调用该函数。由该函数完成对响应的处理。

## 实现管理端的总体步骤

该部分说明了利用SNMP4J编写SNMP管理端的大致过程，读者在阅读之后会对SNMP4J有一个宏观上的认识。在附录部分，作者给出了一个用SNMP4J开发管理站的样例程序，如果有进一步的需要，请参考附录部分。

### 初始化

l 明确SNMP在传输层所使用的协议

一般情况下，我们都使用使用UDP协议作为SNMP的传输层协议，所以我们需要实例化的是一个DefaultUdpTransportMapping接口对象；

l 实例化一个snmp对象

在此过程中，我们需要将1中实例化的DefaultUdpTransportMapping接口的对象作为参数，穿snmp类的构造函数中。

另外，如果实现的SNMPv3协议，我们还需要设置安全机制，添加安全用户等等；

l 监听snmp消息

在此，我们可以调用刚刚实例化的DefaultUdpTransportMapping的接口对象的listen方法，让程序监听snmp消息；

### 构造发送目标

如果实现的是SNMPv3程序，则需要实例化一个UserTarget对象，如果实现的是SNMPv2c或者说SNMPv1，则需要实例化一个CommunityTarget对象。

之后，我们还需要对实例化的对象做一些设置。如果是CommunityTarget的对象，则需要设置版本，重传时间和等待时延。如果是UserTarget对象，我们不仅需要设置版本、重传时间、等待时延，还需要设置安全级别和安全名称。

### 构造发送报文

如果发送的是SNMPv3的报文，我们则需要实例化一个ScopedPDU 类的对象，否则我们需要实例化一个PDU类的对象。之后，我们还需要生成一个OID对象，其中包含了我们所需要获取的SNMP对象在MIB库中的ID。然后我们需要将OID和之前生成的PDU对象或者是ScopedPDU对象绑定，并且设置PDU的报文类型（五种SNMP报文类型之一）。

### 构造响应监听对象（异步模式）

当使用异步模式的时候，我们需要实例化一个实现了ResponseListener

的对象，作为响应消息的监听对象。在构造该对象的过程中，我们需要重写ResponseListener的OnResponse函数，该函数是一个回调函数，用来处理程序收到响应后的一些操作。

### 发送消息

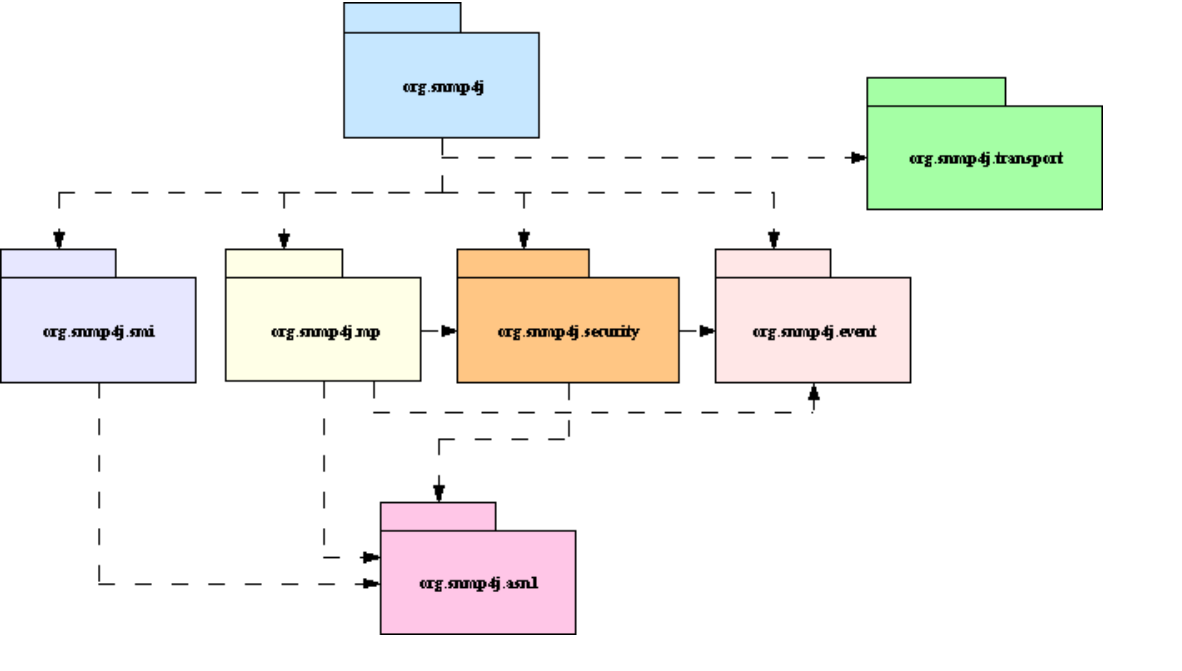
当所有上述操作都设置完毕之后，就可以发送消息了。同步模式和异步模式发送消息调用的函数名字均为send，但是两个函数所需参数不一样。同步模式的参数仅为4.3.2和4.3.3中构造的目标对象和报文对象，而异步模式还需要4.3.4中构造的监听对象。

同步模式发送消息后便等待响应的到达，到达之后会返回一个ResponseEvent对象，该对象中包含了响应的相应信息。

异步模式发送消息之后便会继续执行，当收到响应消息时便会调用监听对象的OnResponse函数。该函数中的语句便是我们对响应的处理

## SNMP4J 源码包中的使用说明

<http://blog.chinaunix.net/uid-540802-id-138732.html>



## Snmp4j 参考

http://blog.chinaunix.net/uid-24948299-id-2538436.html

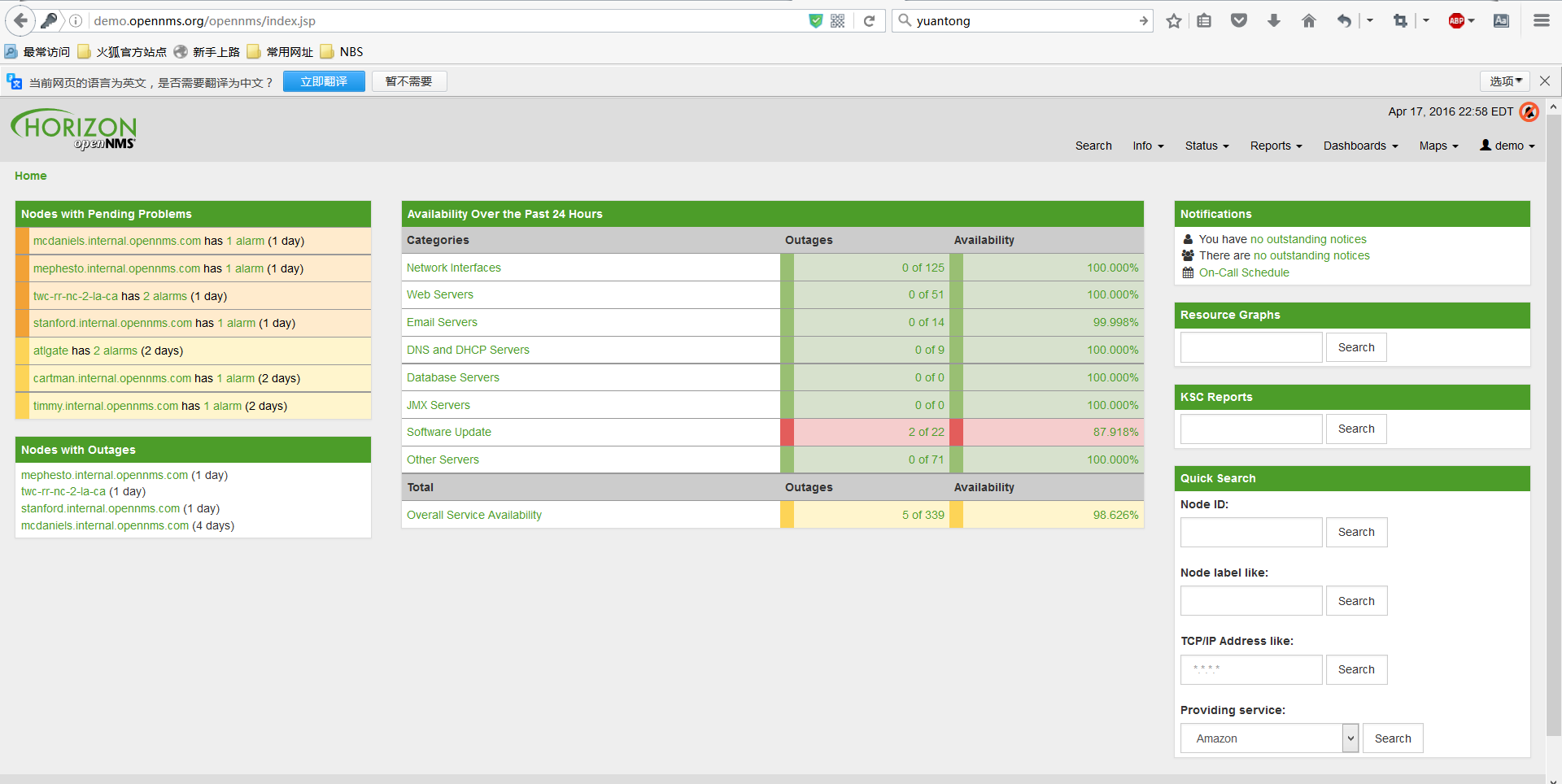
# openNMP

OpenNMS是世界上第一个用开放原始码模式开发的企业级网络管理系统。可用来自动发现网络节点，监控网络服务，如 HTTP，DNS，SSH 等，当系统服务停止时，OpenNMS会依管理者所建立的规则寄出通知，告知运维人员网络障碍，事件汇整，自动执行对应动作，以及服务层级效能监控。 它可以支持SNMP网络管理协议，确保管理的扩展性，并且提供非常灵活的定制功能从而有利于管理范围的伸缩，流量和接点系统硬件使用情况需要在配置 snmp.

在线演示：

<http://demo.opennms.org/opennms/>

* username: demo
* password: demo



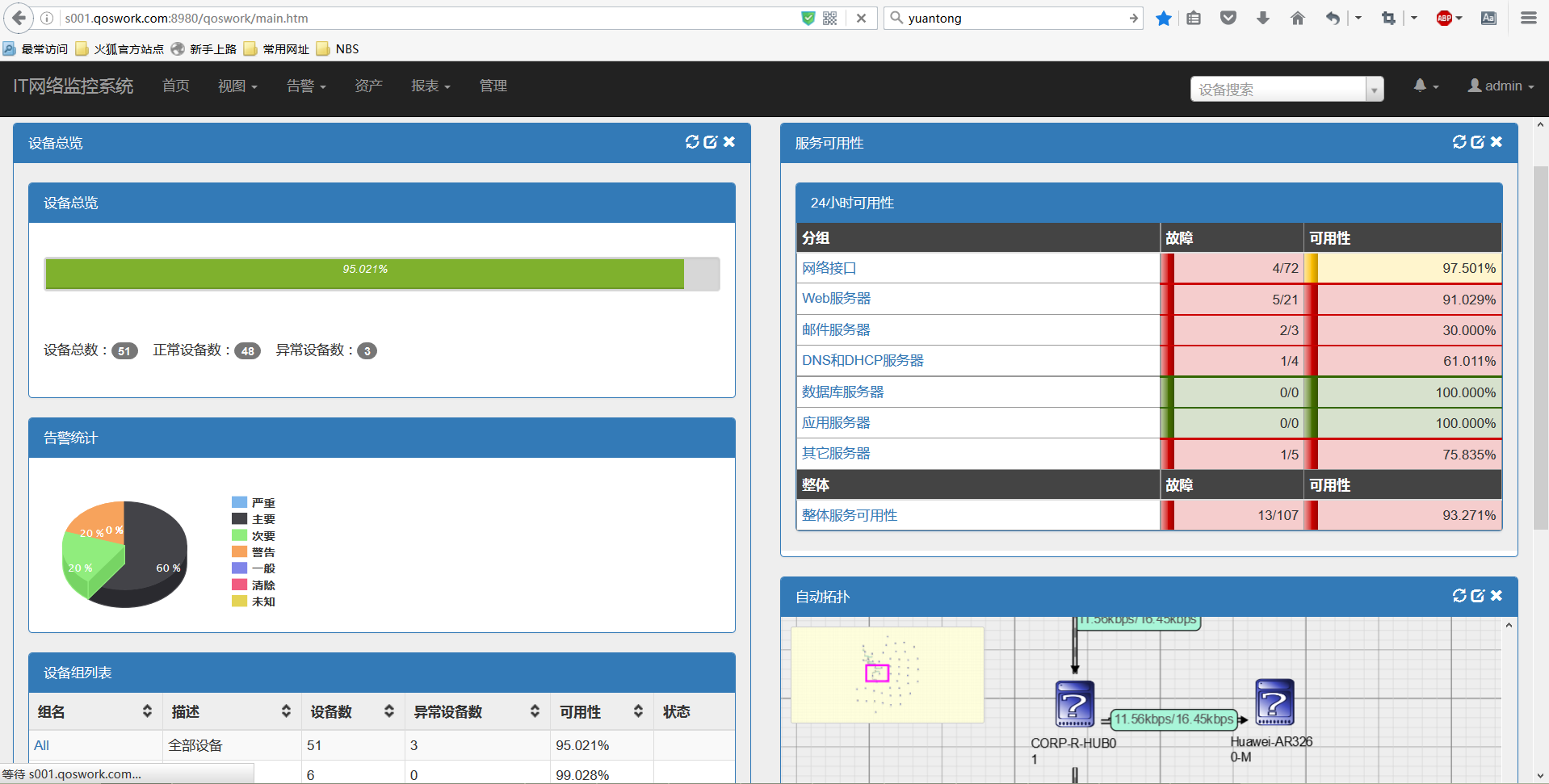
## OpenNMS技术分析与扩展开发

<http://blog.csdn.net/diy8187/article/details/6359255>

## opennms开发

汉化版：<http://blog.chinaunix.net/uid-23549687-id-2398222.html>

demos：<http://demo.qoswork.com>



# 开发步骤

## 深入学习snmp4j.jar

## OpenNMS定制开发（重点&难点）

### 开发环境

OpenNMS基于linux系统，开发时是否要在linux环境下进行

### 数据库

Postgres数据库（是否可换成mysql）

### 开发技术

l OpenNMS采用了xml数据绑定技术（opennms采用的是castor）。根据xml文件的schema定义文件（xsd文件）生成对xml文件到java对象的映射，这样就不需要写解析xml文件的代码而是针对java对象进行操作。因此这些类都是在系统编译过程中由castor包根据xsd文件生成的。(Castor是一个Java开源数据绑定框架，它主要目标是提供Java对象与XML 的绑定，Java到SQL的持久化等.)

l Cleanimports是对java文件中的无用的imports作清理，并通过配置文件提供的格式对imports代码段进行格式整理。

l HttpUnit进行模拟HTTP测试。

l jWebUnit 是基于java的测试网络程序的框架。它提供了一套测试见证和程序导航标准。jWebUnit 有JUnit和HtppUnit实现。

l Junit单元测试

l nekohtml解析HTML，Html Tidy对html 字符串进行修正，并做标准化的处理。

l Avalon主要是一种Server的架构，可以满足配置、日志等服务器程序的需要，

l Jdhcp,java DHCP的实现。

l Xerces解析XML，API与实现有:xmlParserAPIs, xml-apis, xercesImpl

l FOP可以将xml文件转换成pdf，mif，pcl，txt等多种格式以及直接输出到打印机，并且支持使用SVG描述图形。

l jCIFS，用Java开发的SMB客户端库。

l ldap-impl，LDAP java实现。

l smtp.jar pop3.jar，SMTP，POP3协议Java实现。

l JRobin基于LGPL授权的网络性能监控系统，是RRDTool的一个纯Java实现。

l joeSNMP，Java SNMP类库。

所以，要想扩展（Extending）开发OpenNMS需掌握如下技术（Technologies）：

Java 2，XML/XSL，Servlets，JSPs，RDBMS，JMS，Castor，JoesSNMP，RRDTool

# 总结

SNMP是一门很成熟的技术，对简单网络进行管理的协议。这正是我们公司要实现自动运维管理的必不可少的工具。

技术方面有难度，但可以找网络资源进行学习。

业务方面没有接触过，但是正是学习的一个机会。